Exhibit D

revious Next Top Detailed TOC Last Update: 18 janv. 2002

Firewalls: a technical Overview

Table of contents:

Summary: A quick guide to firewalls

- If you just want to connect to the Internet to browse the Web and exchange email for a few users, forget a firewall. Just setup a non-networked PC with simple dialup access to a service provider. Install a simple personal firewall like BlackICE. Unplug the modem when the PC is not in use.
- Solaris/SunOS: Use the instructions in sp/Solaris_hardening.html or sp/Solaris_hardening3.html to strip unnecessary services from the system.
- If you want to use Linux/BSD/IPchains/IPfilter, buy and read [fire3].
- Use proxies *together* with intelligent packet filtering and logging.
- Filters:
  - If you can afford it, buy either SunScreen (or other *transparent* intelligent filter), plus good proxies (for example Gauntlet or FWTK) or a Firewall-1. Even better buy both, to have two different filters - diversity of defence. (This doesn't mean other products are not good, but the author has experience with these products working well).
  - Block at least the following packets: Stop all 69(tftp), 87(link), 111 and 2049(RPC & NFS), 512-514("r" commands), 5151 (lpd), 540 (uucpd), 2000 (OpenWindows) and 6000 (X windows), snmp, exec, bootp, tftp, syslog and all UDP ports except those needed for proxy services (e.g. DNS and NTP). Stop all source routing and IP forwarding packets (i.e. filter out spoofed IP packets).
- For a cheap solution, use two routers + a bastion host running the FWTK (but it will cost skilled manpower to configure correctly).

Even *simple* installations should regularly monitor logs. Alternatively a small UNIX server running the (free) *ip_filter* can be used to offer a more intelligent filter (better) than a router.

- If having the source is important to you, FWTK/Gauntlet may be the only choice.
- Allow only an absolute minimum of services through the Firewall (e.g. WWW, Email, NNTP).
- If incoming UNIX logins are to be allowed, use a secure authentication service such as challenge response (e.g. Watchword) or unique token (e.g. SecurID.), together with SSH.
- Consider regularly scanning your network from the outside, to see (and fix!) what holes are visible with well known scanner tools (e.g. ISS, Satan). Also regularly check the integrity of all files on the DMZ hosts (e.g. via a tool such as tripwire).
- Set up organisational processes for Firewall monitoring, auditing and change management.
- Keeping up to date: make sure that the Firewall administrator is on the *FIRST* or *CERT* distribution list, BUGTRAQ and the Firewalls list too.
- Plan for a disaster! Sooner or later you will be attacked, so have a plan ready (on paper) to reduce the panic induced. If possible set up you own response team on the lines of FIRST.

Introduction

Firewalls are a vast and complex subject, of which a limited overview is presented here. The reader is referred to [fire2] and [fire1] (see references section below) for fine detail. All networks should be divided up into security domains, whereby these domains are isolated from one another by Firewalls, hubs/routers or software (VPNs). Classical Firewalls may be eventually eclipsed by Virtual LANs, in which case Firewalls become those machines that connect two VPNs. Either an entity is required that control access between two domains of different security policies.

This section concentrates heavily on Internet Firewalls, although a Firewall can (and should) be used between any two networks of different security levels / domains.

What is a Firewall?
Definitions :

A *packet filter* stops or allows packets to flow between two networks according to predefined rules. A simple packet filter is a router. It works on the network layer of the OSI model.
A *proxy* is a small, simple program which allows/disallows access to a particular application between networks. It works on the Application layer of the OSI model.
A *Firewall* is a system (or network of systems) specially configured to control traffic between two networks. A firewall can range from a packet filter, to multiple filters, dedicated proxy servers, logging computers, switches, hubs, routers and dedicated servers.
A *gateway* or *bastion* host is a secured computer system that provide access to certain applications. It cleans outgoing traffic, restricts incoming traffic and may also hide the internal configuration from the outside.

Why use a Firewall?

- Each external connection to the internal company network should be secured such that it does not reduce the security of the internal network. Remember that the security of a Network is the security of it's weakest link.
- Every enterprise should have a security policy and connections to external networks should conform to that policy. Normally this is only possible through some kind of firewall.
- A firewall can help stop confidential information for leaving, attackers from entering.
- It can provide detail statistics on communication between the networks (who uses what service & how often, show performance bottlenecks).
- It can provide logging & audit trail of communications, analysis of logs can be used to detect attacks and generate alarms.
- However, a strong firewall doesn't mean that internal host security is not longer needed - on the contrary, most successful attacks come from insiders!
- It is recommended to take a widely used Firewall solution and use it for all external connections.
- Examples of technical threats addresses by firewalls include IP spoofing, ICMP bombing, masquerading and attempts to access weakly configured internal machines.
- Examples of risks reduced by firewalls are attacks from curious & malicious hackers, commercial espionage, accidental disclosure of company data (i.e. customer, employee and corporate data) and denial-of-service attacks.

How does a Firewall protect?
A Firewall normally includes mechanisms for protection at the:

- Network Layer: IP packets are sanitised (source routing disabled, only packets with valid external addresses allowed), and routed according to predefined rules. Some firewalls allow translation of internal IP addresses to valid Internet IP addresses (NAT or Network Address Translation) and other replace all internal addresses with the firewall address (meaning internal hosts cannot be addressed).
- Transport Layer: Access to TCP & UDP ports can be granted/blocked, depending on IP address of both sender and receiver. This allows access control for many TCP services, but doesn't work at all for others (e.g. X11, ftp, portmapper services).
- Application Layer:
  - Proxy servers (also called application gateways) accept requests for a particular application and either further the request to the final destination, or block the request. Ideally proxies should be transparent to the end user. Proxies are stripped-down, reliable versions of standard applications with access control and forwarding built-in.
  - Typical proxies include HTTP (for WWW), telnet, ftp etc. Certain applications such as Internet Email (smtp) are designed for the use of relays or forwarders.

  - The DNS application provides IP address to hostname (or vice versa) lookup. DNS does not really check where information comes from, so it may be possible for an attacker to spoof the DNS service into giving false information, e.g. that the hostname of an attackers machine is that of a trusted host.
  - Applications such as rlogin and NFS use host-names for access control and are hence vulnerable to DNS spoofing.
  - IP addresses should be used on proxy access control lists instead of DNS names (to minimise the risk of DNS spoofing). But even IP addresses can be spoofed if routers are not configured properly and switches are not used.
- Encryption: A firewall may use encryption to provide confidentiality, authenticate or improve integrity. When encryption is used for confidentiality (often called VPNs, Virtual Private Networks), there are two general cases:
  1. Encryption is performed by the firewall, i.e. it is the endpoint of a VPN. The firewall could understand and filter the actual protocol used within the VPN and provide intelligent logging.
  2. Encryption is performed by a host inside the firewall (End-to-End encryption). The firewall sees an encrypted stream but cannot understand it. This is useful is you don't trust the firewall administrator, not so useful if you want to filter the protocols within the VPN. The VPN becomes a point of entry for an attacker that the Firewall administrator cannot detect. Therefore, the VPN end-point inside the firewall must be VERY well configured / monitored and use firewall mechanisms such as strong authentication.
- Dept of defence: A Firewall should also include redundant security barriers, so that a single point of failure cannot compromise the network. The Firewall should be as invisible as possible to users (who could weaken security) and the network (difficult to attack).
- Reliability: Redundant routing, clusters, RAID, cold standbys etc. can all be used to provide varying levels of availability. The reliability of service required should be specified before a firewall is designed.

PROBLEM: Many Internet applications are not "proxy aware" (e.g. Microsoft NetMeeting) and use a combination of ports that can be different for each connection. It is very difficult to allow these protocols to (securely) traverse a firewall. There seems to be a few practical solutions:

- Use intelligent packet filters which "understand" the complexities in the protocol and dynamically open ports when necessary. Disadvantage: requires an intelligent (expensive?) filter.
- On Intranet clients replace the TCP/IP stack with a "proxy" aware stack that diverts calls to hosts outside the Intranet to a specific "proxy server" which then make the connection to the Internet. The proxy server (unless well protected by an intelligent filter) becomes a "sacrificial lamb" that we assume could be penetrated, but has no special access to inside machines and therefore cannot be used to attack internal machines. The Microsoft Proxy uses the above principal to proxy PCs (but not UNIX).
  Disadvantage: requires SW to be installed on all client.
- Buy applications that are SOCKS aware (e.g. Navigator, Applet viewer, etc.), or
  - Try to recompile the application to use the SOCKS proxy system. Disadvantage: need access to sources, or
  - try to replace the system libraries with SOCKSified versions (not easy and prone to error)

Reference Documentation

Ref.	Document number / Author	Title	Date
[fire1]	ISBN 0-201-63357-4 Cheswick / Bellovin	"Firewalls and Internet Security"	1994
[fire2]	ISBN 1-56592-124-0 Chapman / Zwicky	"Building Internet Firewalls", O'Reilly & Associates *Highly recommended.*	1995
[dcom]	Data Communications Magazine (www.data.com)	"Can Firewalls Take the Heat" A test of some Firewalls' performance. *This report has been criticised quite a bit on the Net. It should not be used alone as decision making basis.*	Nov. 1995
[sc1]	SC magazine, page 45	"firewalls" This article is a comparison of the top ten firewalls and good reference for commercial firewalls.	May 1998
[unix1]	1-56592-148-8 Garfinkel & Spafford	"Practical UNIX and Internet Security", Edition 2, ISBN	1996
[corba1]	OMG Document orbos/98-07-03	"Joint Revised Submission CORBA/Firewall Security +Errata"	July 1998
[nworld]	Network World Magazine	"A Flurry of Firewalls" *An interesting summary of current Firewall offerings. Recommended.*	Jan. 1996
[list]	www.access.digex.net/~bdboyle/firewall.vendor.html by Catherine Fulmer	Firewall Product Overview *Contains an up-to-date listing of current firewalls & related product offerings since 1996.*	Jul. 1998
	www.spirit.com/cgi-bin/report.pl	*Firewall Comparison Website*	Jan'00
	www.linuxfirewall.com	Links ti HOWTOS and sources for ipchains, lids, libsafe, NetFilter etc.	Jul'00
[fire3]	ISBN 0-471-35366-3 Sonnenreich / Yates www.wiley.com/compbooks/sonnenreich/ Companion website: www.openlysecure.org	Building Linux and OpenBSD Firewalls *An easy-to-ready book to get you up to speed on ipchains and IPfilter. The companion site includes up-to-date instructions and information (e.g. on OpenBSD 2.7).* ***Recommended*** .	1999

Where to start?

1. Define goals: What services do you want? How much can it cost? Provide a business justification for services.
2. Who/what are you trying to protect, from whom / against what (threats)?
3. What known weaknesses need to be addressed?
4. What risks (likelihood and consequences or impact) do the above threats entail?
5. Develop a strategy to counter the unacceptable threats: policy, organisation, processes and specific technical mechanisms.
6. Select the appropriate technical solution: What tools can provide access to the required services with the specified budget at an acceptable risk? Choose a stable well known technical architecture, test the solution, install it securely.
7. Define a support organisation with roles, processes. You need a well organised team to manage firewalls. Make sure that processes exist for handling a security breach swiftly. Plan for an attack!
8. Submit the firewall to regular monitoring and independent audits.
9. Running a Internet firewall is an endless operation, so it makes sense to follow the technical and social evolution of the Internet.

Policy

You need a written document specify what services are allowed through the firewall and in what direction. Define also whether the default is open or closed i.e. if a service is not specified in the policy, does it mean that it allowed or not?. This must be signed by management, otherwise the Firewall administrator may find himself in deep trouble if security is breached through a hole that "everybody though was covered". An example policy for an Internet Firewall could be as follows:

Internet Firewall Policy (example)

Security Requirements:

1. Access Control

- All Internet access from the corporate network must occur over proxies situated firewall.
- Default configuration: unless otherwise specified, services are forbidden.
- All users are allowed to exchange email with the Internet users.
- R&D department users are allowed to use WWW, ftp and real audio. Others require authorisation.

2. Assurance

- Firewall and proxy machines are to be installed as sensitive hosts. All unnecessary services are to be stopped in the operating system. Users should not be able to logon directly to these machines.
- The firewall policy and configuration must be accurately documented.
- The firewall machines must be subject to regular monitoring and yearly audits.
- Users and firewall administrators should be aware of their responsibilities and be educated so that they can assume these responsibilities.

3. Logging

- Detailed logs must be kept (if possible on a separate server).
- They should be automatically analysed, with critical errors generating alarms.
- Logs should be archived for at least one year.
- The non trivial log entries should be examined daily.
- Statistics of firewall usage should be available.

4. Availability

- The Firewall should offer high availability and fulfill the requirements thereof (backup/restores etc.)
- Processes should exist for change management and incident response.

5. Required Functionality:
Outgoing services:

The following services are required from specific internal hosts (e.g. via proxies) to the Internet:
1. Email, WWW (HTTP), ftp, telnet, SSH
2. DNS (resolve Internet names)
3. News (NNTP)
4. Real Audio

Incoming services:

The following Internet services need to be allowed in by proxy hosts on a specially protected subnet:
1. Email: all users should be able to receive Internet Email, via secured gateways.
2. News (NNTP)
3. Secure Logins (for a small list of people): via SecurID + SSH

Any machines requiring other Internet services will have to be placed on a special outside (insecure) subnet.

They shall be directly on the Internet. Access from the hosts to the internal network follows the same rules as access to Internet hosts.
Services provided to the Internet:

The following Services need to be provided to the Internet (by secured servers in a protected zone):

1. DNS resolution of Firewall/gateway machines.
2. WWW server.
3. Anonymous ftp server.
4. User FTP Server for special projects / collaboration with other companies.

Choosing a type of firewall

1. Buy in or build yourself? : Building yourself is only an option where sufficient skills and resources are available.
Do-it-yourself:

- Testing and verification is more important. Is it right?
- Requires time & expertise. User interfaces are generally better with commercial firewalls.
- Using free products ensure access to source code (although some vendors such as TIS also sell the source).
- Free products may have better or worse **support** (depends on which product). Free stuff that is frequently used (Linux, FWTK, etc.) is frequently discussed in newsgroups and people help each other out alot. Other free stuff has zero support. Some commercial products have support that is so laborious and stupid that support is expensive (time wise) or not capable of solving complex problems (e.g. involving several products).
- Requires investment to maintain knowledge. Will the firewall stay running (securely)?
- Can be useful in large companies where many firewalls of different types are needed by different departments/business units.
- More flexible.

Vendors:

- Ask them what tests they perform, how they guarantee the security & robustness of their product.
- Can be frustrating waiting for the next upgrade to fix current bugs...
- Can be more expensive, but can also be cheaper (easier to install/configure, in general).

2. Cost : Consider the risk, how valuable are the assets being protected? Is an external connection even worth the risk? Establish a budget. Calculate the real cost of the external connection (hardware, software, consultant, support staff, telco charges, etc.). Consider who will pay the costs: the company, departments or individual users. Don't buy a $100'000 firewall to enable 5 users to send email to the Internet! Don't set a budget of $5000 for 500 users!

3. Assurance : Use an OS that has been well tested, ideally a *Trusted* OS (ITSEC evaluated if possible). Remove all unnecessary services and install any security patches.
=> If you buy a ready-made firewall, make sure it uses a secured, stripped down version of an operating system. Ensure that the vendor makes patches available quickly, if needed.

- Assurance can be maximised through *Dept of Defence* (a series of different barriers for protection) and *Diversity of defence* (different methods/products redundantly protecting).
- Ideally all serious Firewall vendors would have their Systems approved to recognised standards (ITSEC or TCSEC). Forget the TCSEC C2 level, it helps little.
- UNIX variants are recommended, since they are well understood and certified versions (B1, B2) are available.
- The NSCA have an approval mechanism for firewalls, that is based on "commercial realities" (meaning it's not as expensive as TCSEC/ITSEC). Check that your firewall is certified.
- Likewise SC magazine (www.westcoast.com) have introduced a certification.
- Harris is well known in the Military and Space industries, it's Cyberguard is the only Firewall approved by ITSEC (May 1998) but the author has had no experience with it.
- NT is interesting, but tends to be complex, requiring many services to run. It also requires reboots when installing /changing configurations and can be difficult to automate.
  NT tends to be used for small sites where UNIX experience is not available.
  Cyberguard, now available on NT as well, includes a module for hardening NT. This should be interesting, but hasn't yet been tested by the author.

4. Risks: What issues should a firewall address?

Technical:

- Single point of failure.
- IP & DNS spoofing, source routing.
- Sendmail, RPC services (e.g. NFS, NIS), old inetd services
- Clear text passwords (telnet, r* commands), bad passwords, default accounts
- Unauthorised access from outside & inside
- Win95/NT file sharing
- WWW server weaknesses
- Anonymous FTP weaknesses

- UDP based services
- inter-host trust
- denial-of-service attacks (UDP, email bombs, SYN flooding, ping of death, ping flooding etc..)
- If backups are needed to guarantee availability,
- Backup Strategy:
  - Avoid a backup server on the internal network. Often, backup services require root access to their clients, potentially opening a weakness in the firewall.
  - or better: Choose a firewall solution that needs as few backups as possible (e. g. read-only hosts, logging onto a dedicated machine - just this machine has to be backed up).

Non-technical:

- Incorrect implementation/administration: clear firewall policy, correct configuration, well defined organisation with processes, roles, responsibility.
- Leakage of sensitive information: Clear information policy with classification

5: Ease of administration

- Access control rules should be easy to implement and control. The rules should be easy to understand e.g. "Disable all telnet connections from Host yyy" instead of "port xxx disabled for inbound connections for IP address yyy with ACK=z, but enable outgoing port xxx to address xxx". A friendly GUI helps a lot.
- Addition of users, hosts or networks should require minimal effort. User and host grouping should be possible.
- Maintenance: if you buy a commercial Firewall consider a maintenance contract to get software updates and fixes but also to guarantee the system availability. If commercial maintenance is too expensive, at least disk mirroring or cold or warm standby is recommended for busy firewalls.
- How can content servers (WWW, ftp) be securely updated from the inside? How can they make queries to the inside with a minimum of risk? One solution is to use the "scp" program that is part of SSH.
- Time Based Access Control: Some Firewalls (e.g. Firewall-1 V3.0) allow you to specify when particular services are available. This is a very useful feature. E.g. perhaps incoming, authorised logins should only be allowed during office hours? The authors knows of no firewalls that allow an *expiry date* to be set for rules, although it would be very useful.

6. Logging

- Enable logging (to write-once media if possible) and monitor logs daily. Although is it easy to log too much, disk space is relatively cheap and to much detail is better than too little.
- Syslog: Log to a machine inside the firewall, whose syslog is not accessible from the outside. Internet servers (on the outside) will have to log locally in this case.
- Set up automatic log analysis and alarm mechanisms (e.g. via Network Management Software, swatch or customised scripts).
- Different levels of logging should be available for each ruleset and notification (via email, pager, audible alarm etc.) of customisable events should be possible. Both dropped, refused and accepted packets should be able to be logged.
- Important and critical events should be summarised and be quickly available for checking.
- Detailed statistics of firewall usage and attack attempts is highly desirable. These statistics show who is using the firewall and how many potential break-ins have been blocked by the firewall. This second item may be difficult, as it requires analysis on the IP, TCP and applications levels, but allows justification to management. Of course *successful attacks* are another thing.
- Logging and alarm facilities differ greatly between different products. Detailed logging, summary statistics, automatic log pruning & forwarding together with a customisable alarm mechanism are required for most sites. If possible, reports should be produced in a standard format such as HTML (easy to read on any platform).
- Precise alarming is very important. It is important to know about major breaches within minutes, but small fiddly problems that are not a major security risk should not be waking administrators in the middle of the night..
- If possible a log browser should be available in GUI and command line form.

7. Organisation: Firewall security is a moving target. Processes should exist to ensure that the firewall administrators are up to date on the latest security issues.

8. Availability: Reliability of service requirements should be specified for the Firewall. Is the connection protected by the firewall mission critical? What down time, when is acceptable? Some form of system backup and/or redundancy will probably be necessary.

Building blocks

Packet Filtering: Routers can be configured as primitive packet filters. However, they have limited (or no) logging capabilities, frequently don't offer state based filtering, often leave high ports (e.g. databases) completely open and have very complex rule sets.
For these reasons, it is preferably to use an intelligent, state based filter or a dedicated host with packet filter software. Note, however, that not all Firewall kits offer state based filtering. If a router is used for filtering, choose one with easy to configure rule-sets. A typical example of filtering required is as follows (finer detail is to be found in the services section below).

- Stop all UDP ports except those needed for proxy services (e.g. DNS and NTP).
- Stop all source routing and IP forwarding packets (i.e. filter out spoofed IP packets).
- Check all incoming packets for address spoofing (i.e. there should never be packets with addresses from internal hosts on the external interface).
- Stop at least the following ports: 69(tftp), 87(link), 111 and 2049(RPC & NFS), 512-514("r" commands), 5151 (lpd), 540 (uucpd), 2000 (OpenWindows) and 6000 (X windows).

- Cisco: Use Cisco firmware 9.21 or later. Examples were Cisco router filters are available at ftp.cisco.com:/pub/acl-examples.tar.Z but seem to have disappeared. Try searching the Web or ftp.cisco.com
- Sample products are listed below in the sample products section.

Content Filtering Many newer firewalls analyse the content of information passing through in order to restrict the information flow according to an information policy. Email, news messages, ftp and http file uploads and downloads can be analysed. Examples of content that can be recognised and (depending on policy) rejected are:

- Email spam
- Viruses
- Java applets
- Trojans
- ActiveX programs
- JavaScript
- Specific file names
- Types of files (e.g. executables)
- Products are listed in the sample products section.

See also securityportal.com/direct.cgi?/closet/closet20000412.html for a discussion on how useful content filters really are.

Proxies: Application proxies (or application gateways) normally offer logging and access control at the application layer, but at the cost of performance (all traffic must pass via the proxy) and complexity (the proxy needs to run on a special host). It may also be necessary to modify the client program. Ideally proxies and filtering should be used together to maximise security. See also the services section below.

- use proxies for all services that must pass between the inside & outside networks.
- configure all proxies such that access to the proxy from the outside is forbidden, except for where strong authentication mechanisms are in place, or for email & news.
- Use IP addresses rather than subnet/host names in access control lists.
- If a proxy doesn't exist for a particular host, perhaps the *plug-gw* in the FWTK, SOCKS, or even the Microsoft Proxy can be used.
- Sample products are listed below in the sample products section.

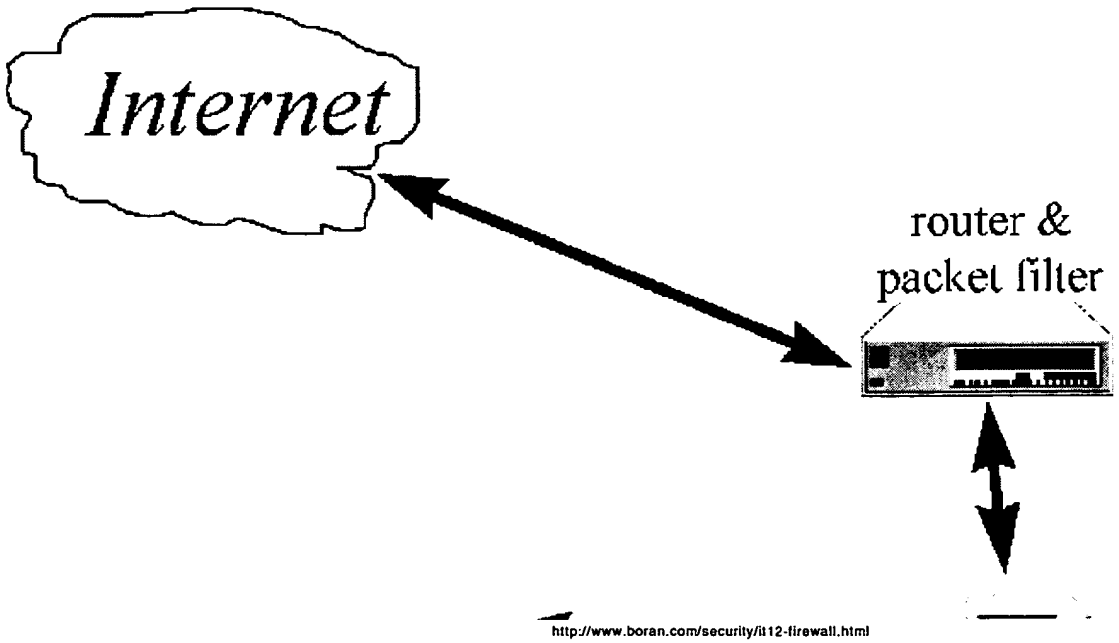
Sample Architectures

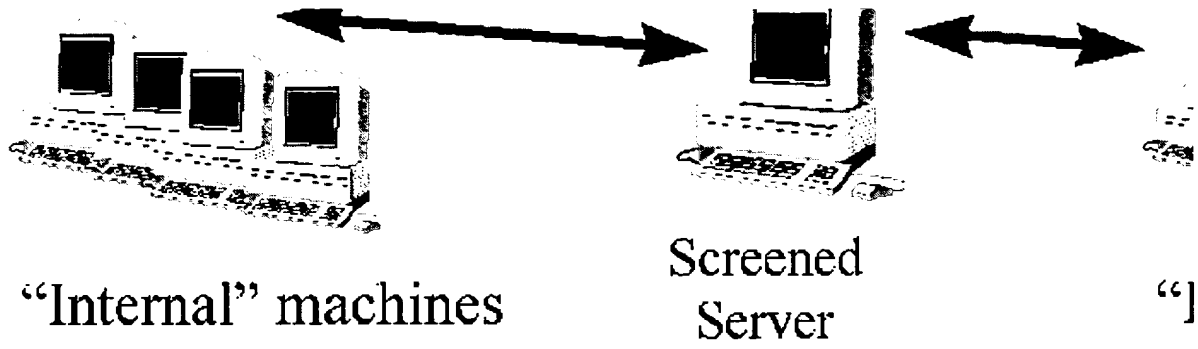
The are many possible ways to set up a Firewall. Here the principle methods are shown. The choice of Firewall depends on cost, performance, availability needs and the sensitivity of the information being protected by the firewall. Highly secure, high performance, high availability systems are not cheap. If high availability is important, it could double costs.

Security can be maximised through *Dept of Defence* (a series of different barriers for protection) and *Diversity of defence* (different methods/products redundantly protecting).

Basic Filter Architecture (screening router)

The cheapest (and least secure) setup involves using a router (which can filter inbound and outbound packets on each interface) to screen access to one (or more) internal servers. A router is normally needed anyway to connect to the Internet, so the filter is for free. This server is the starting point for all outside connections. Internal clients who wish to access the outside do so via this screened server.



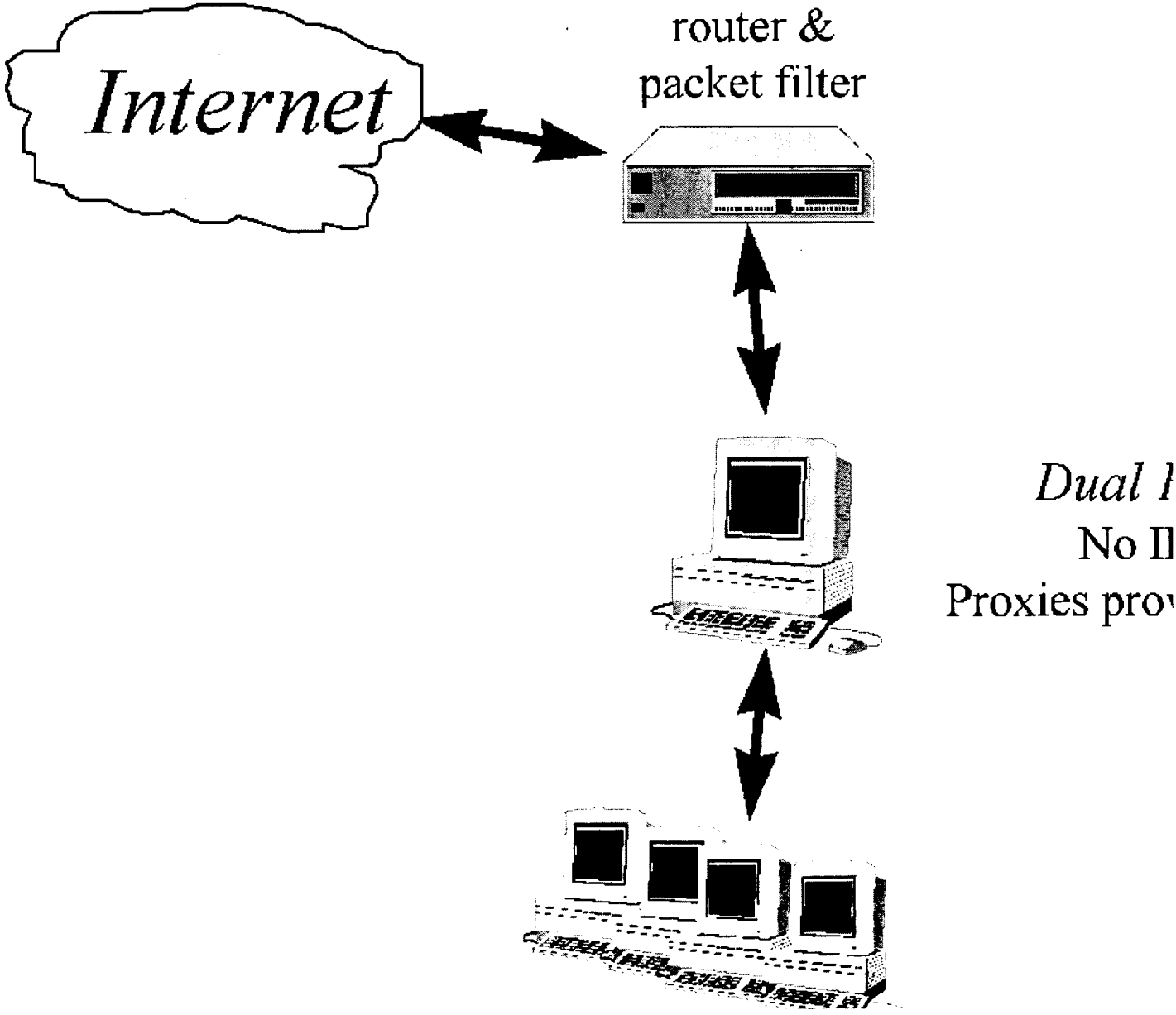


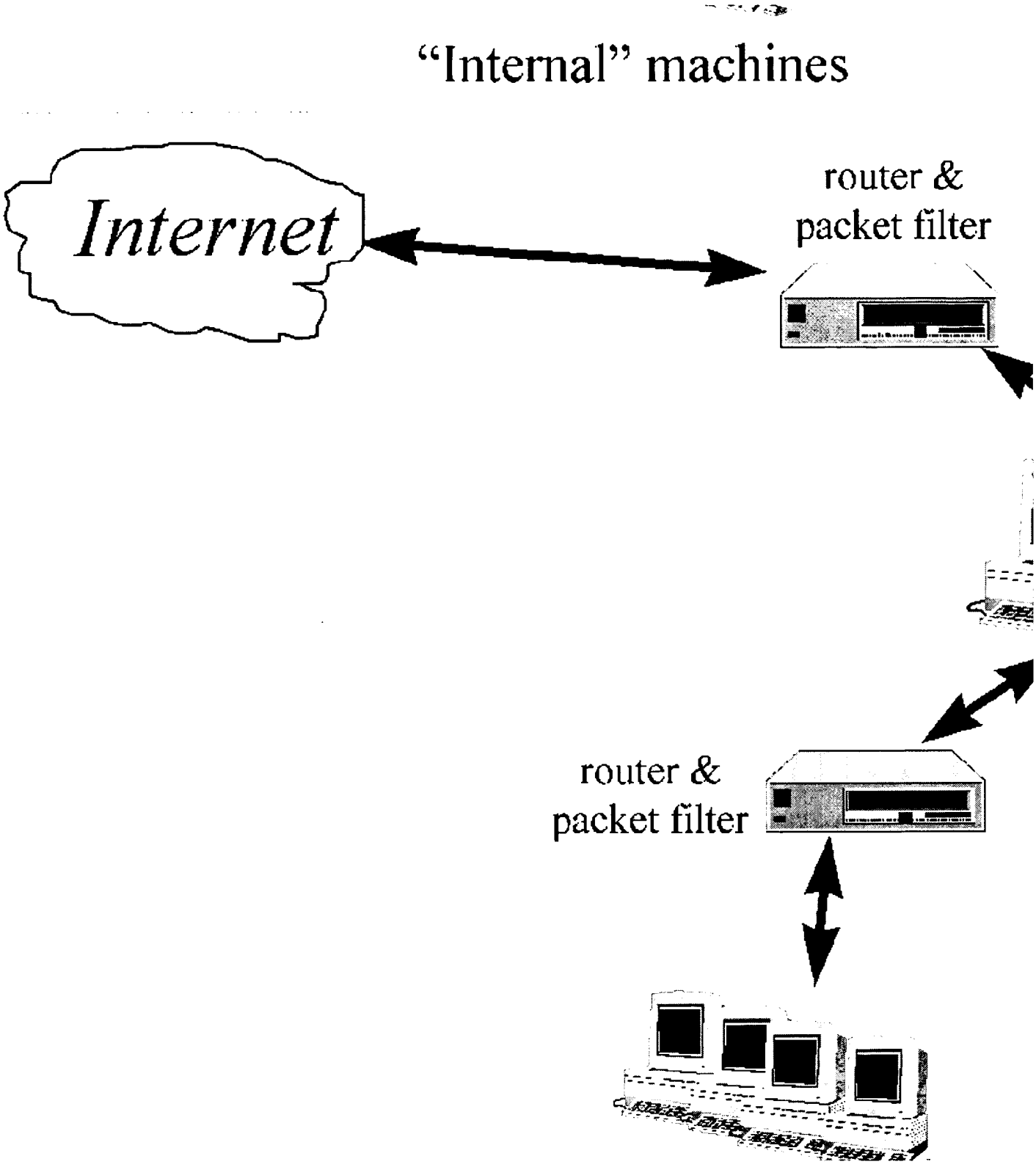
with logging (although such logging is at a low level, making it difficult to interpret).

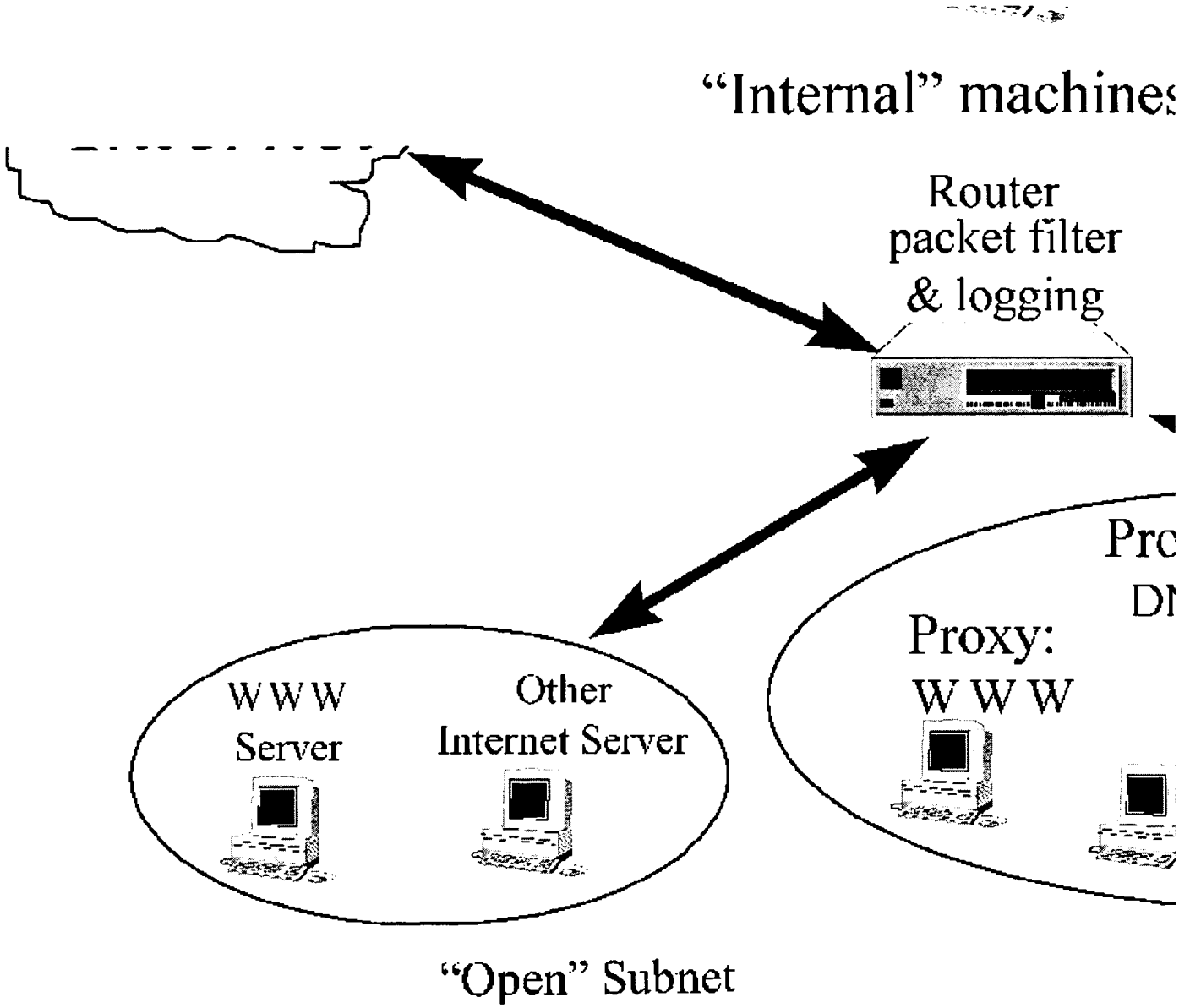
Analysis: This architecture is not recommended, except where finance is a *severe* problem (even then, is it really worth the risk?). As an improvement, an "intelligent filter" (see below) could be used to replace the router filter.

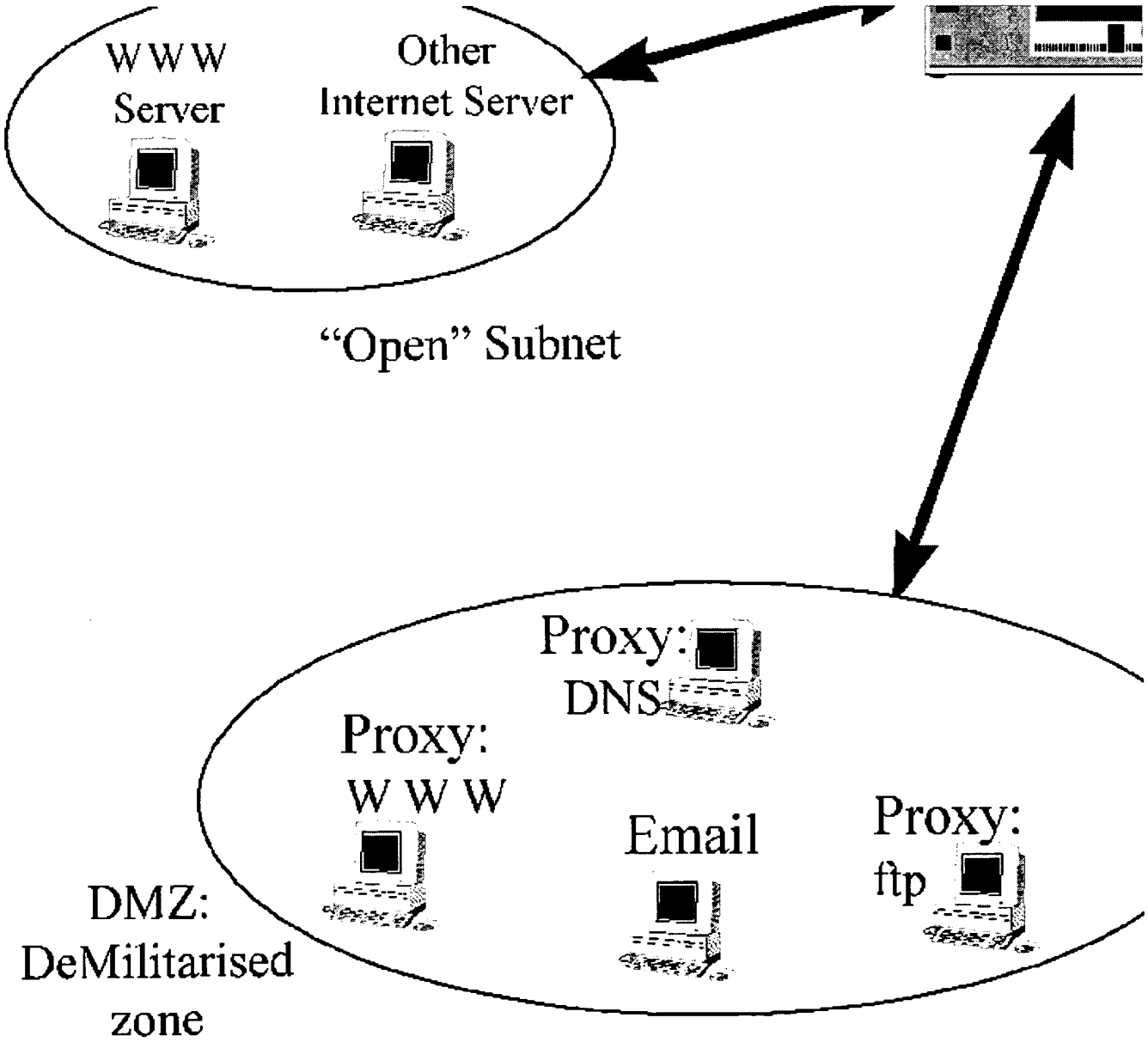
Dual Homed Firewall Architecture

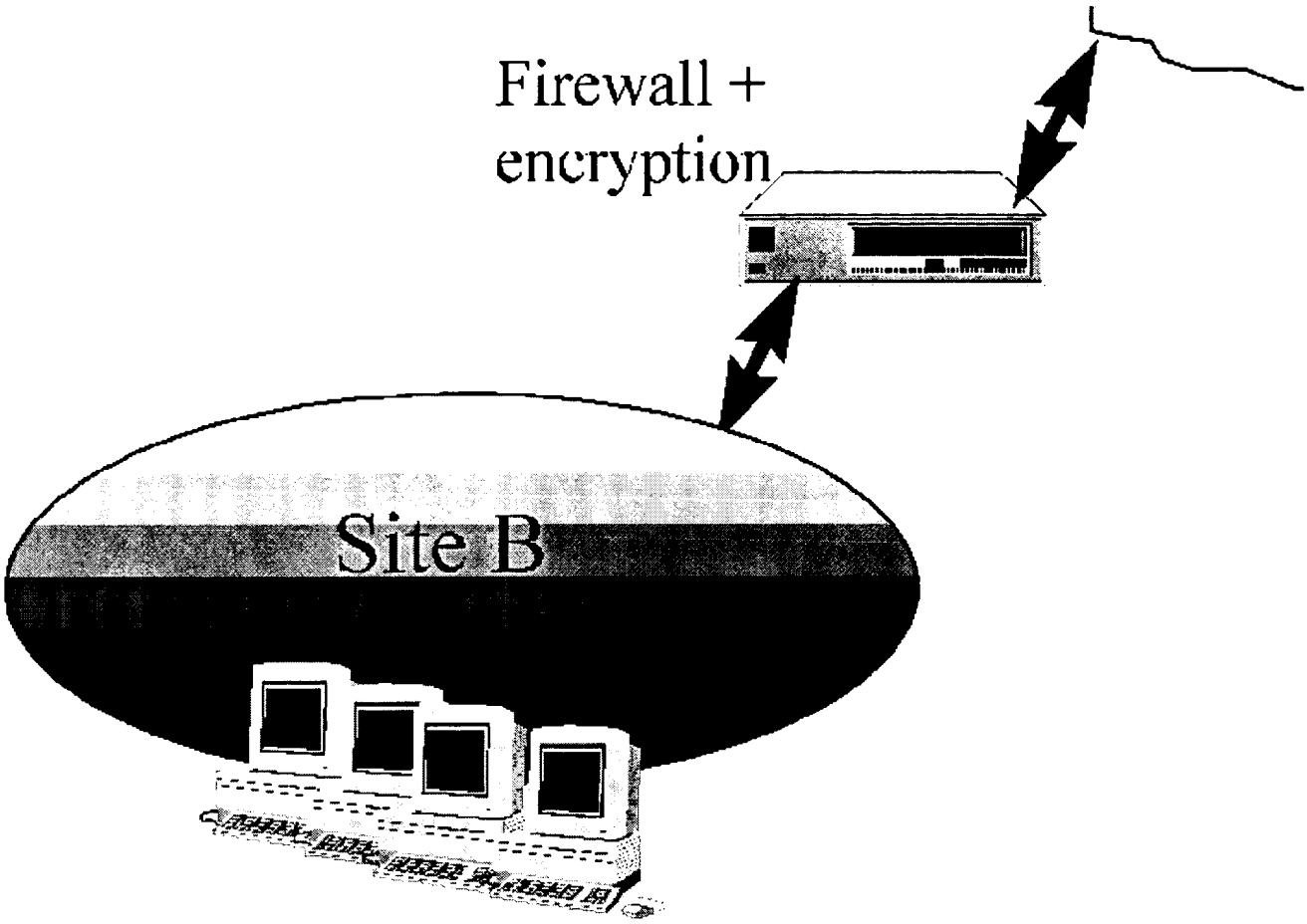
In this classical firewall architecture, a host is setup with two network interfaces, one connected to the outside, one to the inside. Packet forwarding is disabled on the gateway, information is passed at the application level. The gateway can be reached from both sides, but traffic cannot directly flow across it. Normally, a router is also needed for Internet connection.




"Internal" machines
Internet
router &
packet filter
router &
packet filter


"Internal" machines
Router
packet filter
& logging
Pro
DI
Proxy:
W W W
WWW
Server
Other
Internet Server
"Open" Subnet


WWW
Server
Other
Internet Server
"Open" Subnet
Proxy:
DNS
Proxy:
WWW
Email
Proxy:
ftp
DMZ:
DeMilitarised
zone



- NNTP: Use a dedicated news server if possible. Don't allow automated group creation, if high availability is important. Allow external NNTP connections only from the sites you exchange news with (via the news configuration file and the packet filter). Use packet filtering or proxying to connect trusted external NNTP servers to an internal news server and vice versa. Ensure the latest patches are installed (e.g. INN has several weaknesses).
- HTTP (WWW):
  - *Providing* a WWW Server: Use a dedicated bastion host if possible. Carefully configure access control: could someone upload a programs onto the server and have it executed? Control the external programs the HTTP server can access. Make sure there's no interpreters in the cgi-bin directory. .
  - *Accessing* external Servers: Use a proxy HTTP server for external access - do not allow direct access from the internal network. Configure the proxy such that access from the outside to the proxy is forbidden. Use IP (not DNS) access control for performance. Switch on logging (detailed logging may also adversely affect performance). Don't use password protection on the proxy (user frustration, easy to sniff) and make sure that the proxy_xxx environment variables are not set.
    Packet filtering: . Unfortunately, not all HTTP servers use the default port 80. Allow the proxy outgoing access to ports 80, 81, 800, 8000, 8080, 8888, 7070 (SHTTP) and perhaps others.
  - Configure HTTP clients carefully (e.g. disable ActiveX, JavaScript and possibly Java) and warn users not to reconfigure clients based on external advice. Consider using centralised configured computers like NCs, X-Terminals, Net-Terminals, etc.
- Gopher & WAIS: as for HTTP. Use your WWW client as the gopher client.
- Archie: Don't run an Archie server, teach your users to access Archie via WWW gateways.
- Finger: Limit incoming finger requests to a bastion host. Run a replacement finger daemon on this host. Outgoing fingers may be allowed, but consider replacing the client binary.
- Whois: You don't need to run an externally visible *Whois* server.
- Talk & IRC (Internet Relay Chat): Do not allow *talk* between internal machines and the Internet. If it is absolutely necessary, create a dedicated bastion host in the DMZ to which internal users can telnet and talk to the Internet.
- DNS:
  - Set up an external DNS server on a bastion host for the outside world to access. Do not make HINFO records visible to the outside world.

  - Use an up-to-date BIND implementation (ftp://ftp.isc.org/isc/bind/src/) and double-reverse lookups to avoid spoofing (but at the cost of performance).
  - Consider hiding all internal DNS data and use forwarding and fake records, especially if NAT is used. This doesn't make sense for all sites and it is easy enough to get host information from other sources (ping to networks, mail headers etc.). Splitting DNS is also more expensive, but standard practice on large sites that use NAT.
- Syslog: Do not allow Syslog from the outside. Consider centralising syslog on all firewall machines to a dedicated host (by using *loghost* or manually collecting the logs with *scp*) where logs are regularly pruned, archived and analysed.
- Network Management: Your network managers will probably want full access to all machines, either side of the firewall.
  - Don't allow SNMP across the firewall from the outside (this may require special router configuration).
  - Allow ICMP requests outbound, but limit incoming ICMP requests to "necessary" machines (e.g. your network provider's machines). Allow ICMP echo responses either way. Allow traceroute outbound, but limit incoming requests as with ICMP requests above. Allow only *safe* ICMP message types.
- Routing:
  - Do not allow routing protocols (such as RIP) across the firewall, except perhaps between specified, trusted addresses.
  - Use static routes where possible.
  - *Gated* is interested for controlling who can RIP to you.
- NTP (Network Time Protocol): Consider running NTP purely internally (using a server with a radio clock as a reference for example). If you must run NTP to the outside, use an NTP bastion host as proxy server and limit who can connect via the packet filter.
- rexec, rex, FSP, TFTP, NFS, NIS/YP, lpr, lp: Don't allow in either direction.
- X11: Outside connections are not recommended. If it is necessary, use a proxy server (such as FWTK x-gw) even better, use the encrypted X11 forwarding of SSH.
- Email.
  - SMTP: Sendmail is the biggest nightmare in building firewalls!
  - Use the normal store-and-forward features of SMTP to pass all messages via an *email gateway* . Allow no direct connections between the inside & outside.
  - Use a dedicated host for *sendmail* if possible and remove all services/users and restrict who it can connect to. This host will have to resolve to internal DNS servers.
  - Allow external connections only to the bastion. Allow internal servers to access only the *email gateway* and vice-versa. Use smap/smapd (part of FWTK) or an equivalent in front of sendmail on the bastion. Sendmail is to buggy, large & complex to be allowed direct access to port 25.
  - You may have to hack the rulesets to strip hostnames from the "from" address on outgoing email.
  - Configure two mail gateways, both listed in the MX records for higher availability and performance.
  - Don't allow passwords to be transmitted over the Internet via email.
  - POP: Don't use POP over the Internet (passwords are too easy to sniff).
- FTP:
  - Outgoing connections: should use a proxy, or be allowed only from a few internal hosts protected by a Firewall with state based filtering.
  - Incoming: If incoming is to be allowed, then allow only to the bastion host. If anonymous, writeable ftp is to be allowed, protect the writeable area so it can't be used as a transfer area for third parties.
  - Anonymous servers: Restrict the number of people internally who can put files up for anon ftp. Educate them on the issues involved.
- Telnet:
  - Incoming: Avoid totally or severely restrict incoming telnet sessions. Use special authentication proxy servers (with one-time passwords). Even better, use SSH.
  - Outgoing: connections may be allowed with packet filtering or proxying. To protect data confidentiality of a Telnet session, consider using an encrypted version (e.g. SSH or DES Telnet).
- UUCP: If necessary (UUCP is rarely used nowadays), route UUCP over a bastion host with filtering, otherwise switch it off.
- BSD "r" commands: Allow no incoming connections and only outgoing *rlogin* over a proxy. Data confidentiality has the same problems as with Telnet (see above).
- Real audio: A proxy is available from www.realaudio.com which should work with the FWTK, but the author experienced problems and could not get support. The problem is that both TCP and UDP are used. Several vendors offer proxies. The Mediator-1 real audio proxy works fine, is shareware and cheap www.comnet.com.au/htmls/mediator1.html.
- MBONE: is a multicasting application. No known proxies are available.

CORBA & Firewalls (July 1998)

CORBA/IIOP was not designed to expect a "man in the middle" (i.e. firewall) who controls access and manages connections between client & server. A "IIOP enabled firewall" needs to be able to control outgoing access from a IIOP client to a server and also incoming connections from IIOP client to a server behind the firewall. On the server side, the main difficulty is configuring firewalls to listen for IIOP-bearing connections on all the ports, and destined for all the internal hosts (One of the features of IIOP is that *ports are dynamically assigned* , making it impossible to assign a simple IIOP filter on a particular port number).

CORBA 2.2 is being extended to allow for the introduction of CORBA proxies, but products based on this spec (discussed below) probably won't be available until early 1999 at the earliest. Today, there are two methods of allowing IIOP to traverse firewalls (incoming connections to IIOP servers or ORBs). Outgoing methods are discussed in the CORBA 2.2 section.

1. IONA's Wonderwall (www.iona.ie/products/internet/wonderwall/ and the Sysadmin doc) is the only known proxy available. The Wonderwall "proxifies" the Interoperable Object Reference (IOR) of an object and allows remote invocations on the proxy IOR. It can also work in HTTP tunnelling mode.
   Advantages: Provides logging and access control on the object level. Requires no changes to applications. Messages to Orbix & Orbixweb can be encrypted. Iona are the market leaders.

Disadvantages: The issue of callbacks has not been resolved. Proprietary (not an open standard). An unconfirmed worry is that the Wonderwall only works when the client ORB and the server ORB are from IONA (we need some real world testing)..

2. Visigenic has something called Gatekeeper, but it is merely a gateway that can unpack the tunnelled HTTP and route the request to the right server. They also have a proprietary method of using URLs (www.sys-con.com/java/reviews/visibroker/index.html) instead of IORs.
3. HTTP tunnelling: IIOP message are added to HTTP by the client and intercepted by a quasi-HTTP server on the other side that knows how to unpack and handle IIOP.
   Wonderwall can also act in HTTP tunnel mode.
   Advantages: No changes to firewall filter needed, The UBS homebanking (/www.ubs.com/e/telebanking.html) works with this architecture through AdNovum's software ISIWEB (www.adnovum.ch/ISI/isi.html). ISIWEB uses a modified Apache HTTP proxy.
   Disadvantages: Messages are a IIOP/HTTP mix and are no longer CORBA wire-level compatible. The HTTP proxy is "modified" to do something it was not designed to do, a "backdoor" is created. The additional HTTP handling/translation affects performance.

   Definition of IIOP policy/rules and logging of transaction does not happen in the firewall, the firewall has "no control" over what goes through the tunnel - therefore the modified HTTP proxy show also do CORBA object access logging and access control (it does have access to the ne necessary information. ISIWEB doesn't do this yet. Wonderwall?

CORBA 2.2 Firewall specification:
A new document [corba1] proposes changes to CORBA that would allow firewalls to be integrated into the CORBA model. A GIOP proxy is suggested which is a new element that allows CORBA communication across firewalls and supports call-backs and SSL. Three methods for traversing firewalls are suggested, with the first two being optional, the third mandatory.

1. Using well known TCP ports with a packet filter. Firewalls can normally TCP/IP packets based on port number and IP address. For outgoing access the firewall can simple allow the IIOP ports from selected inside hosts to (selected) outside hosts. Such a "well known" port does not yet exists, but one should be defined in addition to an SSL enabled port. The firewall understands nothing about the IIOP and cannot provide access control to CORBA objects. This solution then is only suitable for outgoing IIOP connections in certain situations (e.g. the firewall does actually allow non proxied connections).
2. Using a SOCKS proxy for outgoing connections. By relinking the libraries of CORBA products with SOCKSified TCP libraries, the connection is routed through the SOCKS proxy (in the firewall) on the network layer. SOCKS is an IETF approved standard.
3. Using a (new) GIOP proxy. The following is an extract (in italics) from the proposal [corba1]. Basically it defines a classical firewall proxy that understands the IIOP application layer and can grant or deny access to inside or outside CORBA objects. it requires changes to the CORBA spec and existing products.
   A GIOP Proxy is an application level firewall that understands GIOP messages and the specific transport level inter-ORB Protocol supported i.e. a TCP GIOP Proxy understands IIOP messages. To establish a connection to a server, a client first sets up a connection to the GIOP Proxy. If the GIOP Proxy is an outbound one, the ORB should be configured with the IOR of the proxy object. If the GIOP Proxy is an inbound one, the server's IOR should contain the IOR of the proxy object on the firewall. There are two styles of connection through a GIOP Proxy: normal and passthrough.
   - *A normal connection is where, from a client perspective, the firewall behaves like a server, and from a server perspective the firewall behaves like a client. The proxy can monitor the GIOP traffic. This gives rise to two security issues. Firstly the client may not trust a GIOP proxy, and hence would not want the proxy to examine the traffic. Secondly, the client and server may be using a particular authentication and/or encryption mechanism that is unknown to the proxy.*
   - *Passthrough : the proxy simply forwards on all GIOP messages it receives to the appropriate party. This recognizes that either the proxy is not capable or is not allowed to examine the traffic. In a pass-through connection, the firewall is not responsible for maintaining the GIOP dialogue on the connection, and it may not issue any GIOP messages of its own (such as replies or close connection). Pass-through connections exhibit similar behaviour to a transport level firewall, but on an object level i.e. once the proxy permits access to a particular object any traffic (following the rules of GIOP interactions) may flow uninterrupted though the proxy .*

The proposal also contains solution for the use of IIOP over SSL, with two scenarios: trusted and untrusted proxies.

1. *Untrusted proxies can forward information from a client in the form of a pass-through connection, i.e. the proxy has no visibility into the encrypted byte stream. This ensures the integrity of the client and server communication but leaves little opportunity for access control. This type of connection restricts the proxy's ability to apply its access control list fully, but it is necessary when either the server or client do not fully trust the proxy.*
2. *Trusted proxies can forward connections using a pass-through connection but also can establish separate connections to the server and provide full access control. This allows the implementation of access control either at the server as in the untrusted case or at the proxy on a per operation basis. All trusted proxies belong to a trust group decided by the target servers.*

   Since all proxies will have access to the IOR of the target object, and the certificate of the client, they can judge whether this client may use a pass-through connection or not. Whether or not a proxy allows or denies permission for a client to use pass-through in any given circumstance is up to the proxy's implementor.

The final element of the proposal suggests a mechanisms for supporting call-backs by using bi-directional GIOP or extending the proxy:

> *To provide a more generic solution in addition to the above, GIOP Proxy objects provide an operation that a client may call. The proxy will generate an IOR with appropriate firewall information in it, that can then be exported to the server. The server can establish a connection to the GIOP Proxy, and send traffic on it. The proxy will re-use the connection it already has with the client in a bi-directional mode to send the GIOP messages to the client.*

Sample Products

This section presents several products, with their advantages and disadvantages as percieved by the author. The most detailed reports are on products that have been extensively tested.

An interesting new firewall, not discussed below, that I've not yet have time to check out is the french *Netasq 100* . What makes it interesting is it's time based rules:

> *Time management: The filtering, Network Address translation (NAT) and URL filtering procedures are structured in configuration files that are triggered at programmable times. You can therefore define different procedures during the opening and closing times of your business. This enhances the security of your organization. No one leaves a door open when it is unnecessary to do so!*

See www.netasq.com

ITSEC Approved Products

The ITSEC (see [itsec] and [itsem]) is described in detail in Appendix C. It is a European alternative to TCSEC (Orange Book) and more complete. ITSEC separates functionality and assurance. There are assurance levels E1 through E6. It defines example functionality classes F-C1, C2, B1, B2, B3 which correspond to the TCSEC classes and the new classes IN, AV, DI, DC and DX which are interesting because they include networking (which is missing from TCSEC)

The following is a list of firewalls that have been certified or are undergoing evaluation by ITSEC. See also www.itsec.gov.uk/

Firewall	Level	Cert. date	Notes
Harris Cyberguard Firewall V2.2.1e	E3	Mar.97	
Cyberguard V4.1 for Unixware	E3	Jan.99	
Cyberguard V4.1 for NT	E3	Jan.9	
Black Hole SecuIT 3.01E2	EAL3	Aug.97	UNIX firewall, evaluated on SunOS Operating Syatem (Modified) V4.1, on a Sun SPARC hardware platform.
Borderware 6.1	EAL4	pending	
Checkpoint Firewall-1 V4.0	E3	Mar.99	Firewall-1, Version 4.0 running on Microsoft NT Version 4.0 with Service Pack 3, Solaris 2.6 and AIX version 4.2.1 and HP-UX Version 10.10
Checkpoint Firewall-1 V4.1	E3	pending	This evaluation addresses the core elements of Firewall-1, but also includes the Graphical User Interface, Remote Management, Authentication, Encryption and LDAP interface for FireWall-1 Version 4.1 running on Microsoft NT Version 4.0 with Service Pack 4, Solaris 2.6, AIX Version 4.3 and HP-UX Version 10.20.
Guantlet Firewall V3.01 NT4	E3	Jun.99	
VCS Firewall V3.0	EAL1	Mar.99	

Intelligent filters (screens)

Intelligent filters are relatively new to the firewall marketplace, but are incorporated into the leading firewalls, such as F1 (see below). In this section, specific, well known filters are presented.

IP Filter

> *IP Filter is a TCP/IP packet filter, suitable for use in a firewall environment. To use, it can either be used as a loadable kernel module or incorporated into your UNIX kernel; use as a loadable kernel module where possible is highly recommended. Scripts are provided to install and patch system files, as required.*

It's author is Darren Reed and it looks promising for adding a low level packet access control layer to firewalls or individual hosts. Note: I've read the doc and the book noted below but not actually tried it out.

Features: IP filter offers filtering of protocol (udp or tcp), IP fragments, ports (and ranges), IP options, TCP flags, ICMP type/code and provides NAT, logging, transparent routing, VLSM (Variable Length Subnet Masks). In addition, redirection of services "transparent proxy" and packet state can be analysed to check that TCP packet ack/sequence numbers are correct.

Advantages: free, powerful, source code, works on many UNIX variants, probably easier to use than ipchains.

Disadvantages: no GUI, requires expert configuration. No definition of address groups is possible, which could make the rules for a firewall protection a large number of networks very complicated. Likewise no definition of groups of protocols is available. The filter engine is not intelligent: it does not understand applications protocols like RPC, FTP which makes spoofing the "keep state" feature a risk.

References:
- A great book to buy is [fire3] and check out the companion website.
- cheops.anu.edu.au/~avalon/ip-filter.html
- ftp://coombs.anu.edu.au/pub/net/ip-filter
- mailing list at majordomo@coombs.anu.edu.au with a subject "subscribe ipfilter".
- IP Filter Based Firewalls HOWTO

- SecurityPortal - Firewalling with IPF
- SecurityFocus Introduction to IP Filter, Introduction to IP Filter Part 2.

Note that when compiled for Solaris, a neat SVR4 package is created for clean installation/de-installation.

ipchains

No definition of address group, which could make the rules for a firewall protection a large number of networks very complicated. Likewise no definition of groups of protocols is available.

If you want to use Linux//ipchains, buy and read [fire3] and check out the companion website.

Firedog/Firemasq

Firemasq is a free firewall, design to run on Linux with Ipchains.

Sinus

An extract from the Sinus homepage www.ifi.unizh.ch/ikm/SINUS/firewall/

> *The SINUS firewall is a free and easy way to protect your network from the daily threats of the Internet. .....*
>
> *Filtering of all header fields in the IP, TCP, UDP, ICMP, IGMP packets.*
> *Intelligent RIP and FTP support.*
> *Easy to understand, text-based configuration.*
> *Graphical management interface for configuration of several firewalls.*
> *Dynamic rules, including counters and time-outs.*
> *Extensive logging, alerting, and counter intelligence.*
> *Prevention of packet and address spoofing - GNU GPL license.*
>
> *To install the software, you need a Linux 2.0.x based system....*
>
> *Although the software has been subject to thorough testing, and has been continuously running without crashes for over 12 months, we are confident someone will eventually unconver A BUG in the software.....*

Interestingly, it also comes with a free implementation of SKIP v1 VPN encryption. ftp.tik.ee.ethz.ch/pub/packages/skip/

SunScreen 100, 200, EFS and EFS3

> *Note: The Author has written a detailed article on the newer Sunscreen EFS 3. See sp/SunscreenEFS.html or www.boran.com/security/sp/SunscreenEFS.html . Below only the older versions are covered.*

SunScreen is a recent product from Sun (see www.sun.com) which allows packet filtering (state based), transparent switching between up to 4 Ethernets and IP level encryption is offered with hosts that understand the SKIP protocol.
It does not route packets (because it does not understand RIP, have a routing table and is not visible to IP hosts). It functions like an ethernet switch and needs to know what IP addresses to expect on what interface.
The (original) Sunscreen 100 was sold as a "black box" with Windows-based GUI for configuration. This progressed to the Sunscreen EFS and now the 200 is being sold as software that can be installed on most Sun/SPARCs and has a Motif & Web based admin GUI.
The administration PC or Web GUI is connected to the "black box" (here after called the Sunscreen) via an encrypted TCP (SKIP) connection.
Price is about $20k.

VPN: The Sunscreen can be used to setup VPNs (virtual private networks) using the SKIP protocol. End-point to end-point (tunnelling) or end-point to Sunscreen (gateway = normal) encryption can take place. In normal mode the sunscreen needs (Sun signed 512 bit for export) certificates for each end point and rules can be added to only allow certain services (encrypted) to certain target IP addresses. In tunnelling mode, the Sunscreen simply lets SKIP (IP socket 79 for SKIP V1 and socket 57 for SKIP V2) pass or not, individual services cannot be passed or stopped since the packet is not decrypted by the Sunscreen. The Sunscreen 100 & EFS only support SKIP V1.

Overview:
In the 100: the OS (a stripped Solaris 2.4, with an encrypted TCP/IP stack running in single user mode) is embedded on a CD-ROM. The hardware is simply a SPARCstation 5 with 1GB disk, 32MB RAM, a floppy drive, a CD-ROM and a quad Ethernet interface. The Sunscreen boots from CD-ROM and uses the internal hard-disk only for swapping, logging and configuration information.

In the 200: The Sunscreen software has two parts, the admin station and the Screen. Both can be installed on most SPARCs with Solaris 2.5 or later. The admin station generates a boot floppy which is used together with the 200 CD-ROM to install the screen with Solaris 2.5.1, configure it and install the screen software. An installation log is written to the diskette (which is very useful for debugging). Several Screens can be managed by one admin station and several admin stations can access the same Screens.

Advantages:

- It can be reinstalled in 15 minutes by simply inserting the configuration diskette and turning it off and on and then connecting from the

admin GUI & download the correct config.
- Transparency: it acts like a bridge and has no IP address, making it difficult to attack.
- It is robust and reliable (except for one bad experience...).
- The filters are stateful and filter most TCP/IP protocols and services. Non IP protocols (such as IPX) can be filtered by Ethernet packet type. A real audio filter is available as a patch for the 100 and is included by default in the 200..
- New user defined services may also be filtered. (but you can't define your own state engines).

Disadvantages:

1. Expensive (IMHO).
2. 100: It is impossible to automate configuration and monitoring, since on a GUI is available (i.e. no command line interface). The 200 is a vast improvement, with a good command line interface suitable for remote administration and script automation.
3. There is no "net meeting" filter.
4. Custom filter engines cannot be created.
5. 100: It is not possible to configure a *range* of ports (**big problem**, depending on your needs) . [fixed on the 200].
6. Source code is not provided.
7. Rules don't have an "expiry date", nor can they be applied at certain times of day, certain days of the week, nor do they have a comment field.
8. 100: No NAT (Network Address Translation) feature. [fixed in the 200].
9. The "normal" export version has small (40bits = not very useful) encryption key length.
10. No proxies are included, it is a pure filter. So you still need to buy proxies for your firewall.
11. Rule conflicts are not allowed, making certain configurations irritating.
12. 100: The GUI is not very intuitive and clumsy at times. Error messages are vague, finding the real source of problems can be very difficult.. It is full of bugs (see below), none of which have been fixed by Sun since April'96. The 200 is better (more stable), but still has a primitive GUI.
13. 100: The administration PC cannot be reinstalled, as no software is delivered (the new html front end might solve this). However, a newly installed PC can "get" the current configuration on a production Sunscreen and use it, so the configuration files are not lost if the PC dies.
14. No high level log analysis is available, with logging occurring on the packet level. The log browser is primitive.
15. Cabling can be problem, since only RJ45 is supported (i.e. not AUI).
16. Bugs (100):
    - 100: "Passed" and "Failed" packets can be logged, but not packets dropped at the interface.
    - Crossed RJ45 can be used between the Sunscreen and the PC, but it's not recommended as it can cause transmission problems.
    - Problems have been experienced with the SNMP udp state based filter.
    - When defining "subnet addresses", if the form XX.XX.XX. is entered, the Admin PC adds a number on the end giving XX.XX.XX.4 (for example), instead of it being interpreted as XX.XX.XX.0. *Workaround* : make sure you add the .0 to the end of subnet addresses.
    - No documentation on the modified PC-NFS is delivered. Problems can be difficult to troubleshoot.
    - The GUI has a serious bug which prevents downloading of Sunscreen configuration. It happens in several situations, e.g. when the administration icon and log browser are used together. Sun have known about this for months, but not fixed it. *Workarounds* : a) restart the PC b) stop the admin utility, delete the *.pfm files in the config directory, or c) upload a configuration before you download one. For initial installations where no configuration can be uploaded, try to download a config with no rules and download the real config afterwards.
    - The traffic statistics cannot be reset to zero, except by rebooting the Sunscreen.
    - Log files greater than 32MB cannot be transferred to the PC every time, often the files are chopped.
    - Logs cannot be automatically downloaded from the Sunscreen, or transferred to another system.
    - The log browser doesn't count the number of packets in a log file correctly.
    - The "summary" log entries don't show the tcp/udp port number concerned.
    - Downloading large configuration to the Sunscreen often breaks down under mysterious conditions (with a direct crossed RJ45 and hub connections).
    - The rules are recompiled before each download, even if they haven't changed. This can make life very slow.
    - The rules compiler gives cryptic error message, meaning that lots of time can be lost with simple rule errors!
    - When creating a Sunscreen install diskette, a "gateway" i.e. default router must be defined, even though this may not be desirable, for instance if you only want to manage the Sunscreen on a local network. This bug increases security exposure.
17. Bugs (200):
    - There is a jumbo patch for the 200, make sure you install it. Crossed RJ45 can be used between the Admin station and the Screen, but if *hme* (100MB interfaces), it refuse to work or be extremely slow. This problem *should* be fixed by the Jumbo patch for the 200.
    - Problems have been noted with "name mangling" of files on the installation floppy. If the installation fails, log on to the Sunscreen console, mount the floppy & check filenames. If there are any with ~ (tilde), they need to be renamed. This problem happen with Solaris 2.6 admin stations (since 2.6 has different DOS name mangling from 2.5)..
18. Recommendations:
    - Change the console password from the default value.
    - Consider installing a "warm standby" for high availability. The "warm standby" is installed using the same installation diskette as the master, after the configuration is downloaded. If the master goes down, just switch the cables to the backup. FTP and other connection oriented sessions will be blocked, but HTTP sessions won't really notice. This type of standby machine is also useful when effecting major configuration changes (you can always switch back to the standby if the configuration is wrong).

Proxies & filters

There are many products on the market today. The following is a sample of the most well known (to the author) or innovative. A detailed, up-to-date list many be found on the net see [list]. See also the intelligent filters section above.

TAMU drawbridge (free)

Drawbridge is a public domain (ftp://ftp.tamu.edu), Texas University was originally a PC (DOS) based packet filter (not stateful).

Drawbridge is still regularly updated, and is now based on FreeBSD. http://drawbridge.tamu.edu. Drawbridge's main strength is that it's performance remains constant whether it is processing 1 rule or 1000, very different from most firewalls where performance is directly related to the number of rules processed.

TIS Gauntlet & FWTK (commercial/free)

FWTK (Firewall Toolkit) is *the* set of free utilities for building your own proxies, Gauntlet is a fully fledged commercial version of the same. FWTK is often used to complement Vendors firewalls who lack certain features/services.

General

- Full source code is provided. Utilities are quite small, meaning that source code verification and modification is not that difficult.
- FWTK is not a unified program, but a collection of independent minimal utilities which work together, configured in one configuration file. No GUI is available. This makes the individual tools very flexible and easy to use "alone", but is more difficult to monitor and configure for unskilled personnel.
- Proxies for ftp, telnet, rlogin, http, and (sort of) SMTP & NNTP are provided, along with an authentication server which interfaces to the well known authentication systems (S/Key, SecurID, Kerberos...).
- Inetd services may have access control added on an IP address basis with the *netacl* wrapper.
- All proxies *chroot* and can be run under a given UID for added protection.
- The HTTP proxy does not have caching (caching can improve performance enormously), nor does it support SSL.

Free version: FWTK (Firewall Toolkit)

- A reduced version (no packet filtering or statistics analysis) is available for free (see www.fwtk.org or ftp://ftp.tis.com/pub/firewalls/toolkit/fwtk.tar.Z , current version is V2.1), which has ensured that it is the most popular firewall in use today. Before downloading the free version, you need to register with TIS.
- The public domain version provides features not found in some commercial products, such as an X11-over-telnet proxy (SSH does this even better).
- The *smap* utility for protecting access to sendmail works well and is recommended.
- There is no script to clean up the operating system for you. You need to strip the OS to the bare minimum manually.
- FWTK also includes some useful utilities such as reporting tools (ftp-summ.ch, http-summ.sh, tn-gw-summ.sh, weekly-report.sh), portscan and netscan.

Commercial version (June 99)

- Packet filters are not stateful and apparently has some problems with udp.
- The GUI is not up to the standard of market leaders such as F-1. It needs getting used to and could allow mistakes to be made.
+ A command line tool is provided in addition to the GUI
+ TIS also sell a trusted UNIX, but it is not included with Gauntlet.
+ Tools for detailed statistics analyses are provided.

SOCKS 5 (21 Oct.1996)

SOCKS is a generic proxy system that can be compiled into a client side application to make it work through a firewall. It is easy to use, but does not additional support authentication hooks or extra logging. See ftp.nec.com/pub/security/socks.csts/socks5/ . Netscape's HTTP proxy and the Java Appletviewer support SOCKS. It is now an Internet standard approved by the IETF.

The main problem with SOCKS is that the clients must be "SOCKSified", but there are quite a few SOCKSified clients and appropriate libraries are available for some platforms.

The following is an extract from the Socks version 5 home page, www.socks.nec.com :

> The SOCKS5 protocol, also known as authenticated firewall traversal (AFT) is an open Internet standard (rfc1928) for performing network proxies at the transport layer. It is intended to resolve several issues that SOCKS4 didn't address fully or omitted completely:
>
> Strong authentication
> Authentication method negotiation
> Message integrity and privacy
> Support for UDP applications
>
> There are two additional SOCKS5 related standards for supporting two authentication methods. One is the "Username/Password authentication for SOCKS V5" (rfc1929). The other is the "GSS-API Authentication for SOCKS V5" (rfc1961). Besides providing the authentication, the GSS-API (Generic Security Service Application Programming Interface) also supports message integrity and confidentiality.

The public version available from NEC compiles easily on most platforms and offers proxies for ping, ftp, traceroute and telnet out of the box.

It can also be used to setup secure tunnels (VPNs), but this feature has not yet been tested by the author. This principal problem with SOCKS is that the client must be *SOCKSified* i.e. know how to talk to the SOCKS proxy server. This modifications are minor in most cases but do require compilation. However in many cases (Winsock, Solaris), the system shared libraries can be replaced with SOCKS aware libraries allowing the client to use SOCKS without being recompiled or changed!

Author's experience:

- It works well on Solaris and is recommended, but difficulties have been noted with Win95/NT (i.e. replacing TCP dlls with SOCKSified DLLs)..
- SSH (on Solaris) works perfectly with SOCKS (if compiled with the correct option).
- Recommendation: Make sure that the SOCKS server is configured to refuse access to clients outside your network. It has happened that a badly configure SOCKS proxy offered proxied connections to the inside!

Firewall-1 V3.0 (Jan.1998)

Developed by Checkpoint Technologies and also resold by Sun, this product offers both **state based packet filtering** and proxy servers. It is one of the most popular Firewalls. Performance is very good [dcom] and is a favourite for some [nworld]. Good GUI for filter configuration. Source code not provided. The version sold by Sun is often 6 months behind that sold directly by Checkpoint, but it is better integrated into the Sun environment. **Recommended**. See www.checkpoint.com

- The packet filter works by hooking directly into the OS TCP/IP drivers.
- It has to be one of the most feature packed firewalls with NAT, VPNs, stateful inspection, a customisable filter engine, very flexible rule configuration, time based access control, separate "system policy/properties", filter engine for new protocols such as VDOlive, NetMeeting etc., filtering of content (viruses, applets) and so on. In fact it may be to complex for some!
- It's price isn't too bad, starting at about $3000 for 25 hosts.
- Runs on UNIX and NT.
- It is possible to highly automate configuration and monitoring.
- There is no script which cleans up the operating system for you. You need to strip the OS to the bare minimum manually.
- Bad:
    - You can use the command line or GUI, but not both!
    - The GUI, while very pretty can be highly unstable (core dumps).
    - Buggy....

Cyberguard Firewall

This Firewall contains both packet filters (stateful?) and proxy servers. Excellent performance [dcom]. Based on Harris's Trusted UNIX. Cyberguard is well known in the Military and Space industries. Runs on NT and UNIX (SCO). This Firewall sounds very good and is the only one approved by ITSEC (July 1998) but the author has had no experience with it.

Norman Firewall (June 1996)

Norman Data Systems is another military supplier, offering a Firewall based an a B1 (TCSEC approved) operating system, HP-UX 10.09.01 CMW or Secureware SMP+ 2.4 (an SCO derivative). This Firewall has some unusual features:

- File transfers can be automatically searched for viruses signatures or "hot words". The transfer is stopped and the files stored on the Firewall for the administrator to examine.
- support for B1 level labelling.
- dual-homed, proxy only (http, ftp, telnet..) gateway. Direct packet filtering between inside and the outside is not supported.
  See www.norman.com .

"black box" firewalls

- The *Sonicwall* from Sonic Systems Inc www.sonicsys.com is an interesting little hardware device. It looks like a small black router, but is in fact a little firewall with NAT & stated based filtering that can also defend against SYN flooding, Ping of death, IP spoofing and filter ActiveX, Java and cookies. Configured via a Web browser. Cost ~$3000.
- *GNAT Box* www.gnatbox.com is PC-based hardware solution that can be controlled remotely from UNIX or Win95/NT and seems to offer many of the features required in a new firewall.Filtering does not seem to be state based.
  A scaled down free version *GNAT Box light* is also available, than can be used for a small number of users.

Windows NT/2000 Proxies

Here's a quick list, that needs completion....

- AnalogX proxy (free). *AnalogX Proxy is a small and simple server that allows any other machine on your local network to route it's requests through a central machine. So what does that mean in English? Simple, run Proxy on the machine with the internet connection; configure the other machines to use a proxy (it's very easy, there's a detailed description in the readme), and voila! You're surfing the web from any other machine on your network! Supports HTTP (web), HTTPS (secure web), POP3 (receive mail), SMTP (send mail), NNTP (newsgroups), FTP (file transfer), and Socks4/4a and partial Socks5 (no UDP) protocols! It works great with Internet Explorer, Netscape, AOL, AOL Instant Messenger, Microsoft Messenger, and many more!*
- Microsoft Proxy
- Mail Essentials
-

HTTP proxies

For HTTP, a proxy with caching will boost performance significantly.

- Apache, the most popular web server (www.apache.org) is always worth a look.
- The W3C HTTP caching proxy is know to perform well, but seems to have some problems with the ftp proxy. See www.w3c.org . The command line interface may make it difficult to configure for some.
- The (commercial) Netscape proxy is well known, a bit expensive, but seems to work well and it has a HTML GUI interface for configuration, together with support for SSL and HTTPS. The GUI configuration interface is accessed via an URL and an administrator password - so make sure you choose a good password! It is recommended to restrict access to the admin URL (see obj.conf). The administrator name & (encrypted) password are stored in a file (admpw), so make sure that this file is only readable/writeable by the user under which Netscape runs (e.g. http) or root. Make sure that the user under which Netscape runs has a secured (*read* securely configured and if possible blocked) account. If the encrypted password is deleted from the file, there is no longer any password.
- The FWTK http proxy has no caching and some stability problems. The stability issue has (apparently) been addressed in the commercial version.
- The Microsoft proxy (runs only on NT) is interesting but not as feature rich as Netscape and difficult to use for high volume sites. It does however allow proxying of difficult protocols such as NetMeeting.

Content filters

Analysis of information flowing through firewalls and restrictions based on content can help to implement information security policies. An article discussing content filter products is on page 50 of the August 1998 issue of SC magazine (www.westcoast.com).

Most commercial firewalls offer content filtering, or hooks to attach content filtering to their proxies. In addition some traffic monitoring tools for intrusion detection also do a type of content filtering.

- *MIMEsweeper* and *WEBsweeper* by Content Technologies (www.mimesweeper.com or www.contentsecurity.com) are probably some of the best known filters. They run on a dedicated NT box, installed as a mailgateway, respectively http proxy within or in front of the firewall. Cost ~$3000 for 100 users. In late 1998, *Secretsweeper* was released which allows scanning of attachments encrypted with *Secrets for Windows* . Apparently a similar product for scanning S/MIME encrypted attachments is also in the works.
- *SessionWall-3* from AbirNet Inc. www.abirnet.com, cost ~$1600 can use used to setup a detailed information policy that also include the sizes of files that may be downloaded at what times -can help to reduce peak network loads.
- *TFS Gateway* from Tenfour www.tenfour.com offer a mailgateway for NT with lots of features like spam filtering, virus scanning (3rd party), encryption (with PGP) and message tracking. Designed for small and medium size organisations. Cost ~$600.
- The *Finjan* www.finjan.com filter is unique in it's fine grained filtering of Java programs/applets.
- Checkpoint have define the open OPSEC standard (see www.opsec.com) which should allow security products to integrate with one another. At the moment, this means that several third party products are available which integrate cleanly into the Firewall-1, in the areas on Content analysis, authentication, intrusion detection, event analysis and availability.
  Definitely interesting if you use F-1s....
- The *PGP Policy Management Agent for SMTP* is a tool that works together to the Email gateway to ensure that email messages containing PGP passing through the gateway conform a policy. See also
  Features: *Rejects all encrypted messages which have not also been encrypted to one or more corporate message recovery keys. Allows, Disallows, or Requires the use of digital signatures on messages. Determines whether all email and attachments must be encrypted before passing policy requirements. Allows the rejection of all messages that have been encrypted with only conventional encryption. Conventional (also known as symmetric) encryption requires the recipient to have the same passphrase as the sender. Disallows the use of encryption to specific keys. Limits the checking of policies to be enforced for a specified IP address or domain.*
  www.pgpinternational.com/product/pol-man.html
  Policies can be designated for an entire domain, network, subnet, or IP address.
- Common Content Inspection API (see www.news.com/News/Item/0,4,26737,00.html?owv):
  - *The API is designed primarily so that "content screening" software like antivirus or software to block malicious Java applets and hostile ActiveX controls can work with firewalls, routers, proxy servers, and caching devices--so-called "perimeter" products that sit on the edge of a corporate network.*
  - *The effort originated with Stardust and Finjan, whose software block malicious applets. A first draft of the scope and goals of the effort, due October 15 1998, is being written by representatives of Finjan, firewall leader Check Point, antivirus vendor Symantec, and virtual private networking firm Aventail .*
  - *Bradley Brown, Check Point director of business development, said the new effort could be a successor to CVP "content vector protocol" , which already has wide adoption among content-screening software vendors.*

Log analysis

Finding ways to analyse your firewall and syslog logs for attacks, unusual behaviour or for statistics is rarely easy. Most commercial firewalls provide some kind of Log browser, that it often quite primitive.

- If possible get a firewall that has both a GUI and command line log browser. The former will help to quickly learn the system, the latter for making you own (perl) scripts for automated log analysis.
- Logs should be scanned regularly for unusual entries. If possible logs should be centralised and written to write-once media.
  Firewall filters: use of fail rules, invalid / spoofed IP packets, traffic/service/user/item based usage.

General Log Analysis Tools: Automatic analysis of logs and alarming can be done via:

- **DIY (Do It Yourself)**:
  - If you want concise reports of only the unusual stuff, you can't beat writing your own scripts and throwing away what you know is OK, calculating statistics for normal usage and reporting the remaining (presumed unusual) entries. This method will also ensure that system problems are detected quickly.
  - When it needs to be done for several machines, either use SSH or write a client/ server socket program that runs local analysis and delivers the results to a central console.
  - Perl is the favourite tool because it is so powerful, works on most systems and Perl modules exist for analysing the NT event log, WWW logs, UNIX (syslog) logs and writing client/server sockets easily.....
  - Try *grep* , *sort* & *cut* if you find perl difficult..
  - Things to check for on firewall filters: use of fail rules, invalid / spoofed IP packets, traffic/service/user/item based usage.
- "Default" tools are delivered with many Systems. When buying new systems, try and ensure there is a GUI and command-line log browser included.
  Examples: the F-1 *(fw log* and *fw logexport* and GUI browser), Sunscreen log browser/command line, NT event viewer & resource kit tools, etc.
- Darren Reed, the author of IP Filter, announced the release of *Nsyslog* in June 1999, a syslog implementation that supports TCP connections and can be used with SSL to encrypt delivery of syslog messages. Sounds interesting. but not yet tested.
  coombs.anu.edu.au/~avalon/nsyslog.html
- **Webtrends for Firewalls** V1.1 www.webtrends.com
  Verdict: couldn't get it working!
  - Can analyse F-1 (via OPSEC or exported logs), Raptor, Gauntlet UNIX/NT, Cisco PIX, Lucent, Secure Computing, AltaVista logs. But requires direct access to these files.
  - Incoming & outgoing web requests can be analysed.
  - It can report in French, German or English and deliver reports via email, ftp or in real time to HTML files.
  - Report format is word, excel, CSV, ASCII or HTML. Report layout & content can be customised.
  - It runs on NT (as a service optionally) and comes with a friendly GUI.
  - Not *too* expensive at $999.

  Disadvantages:
  - Could not get it working (with F-1 exported logs).
  - GUI not available on UNIX?

- **Axent Omniguard ITA** (Intruder Alert) is a system for gathering logs and monitoring changes on an array of sensitive hosts, responding to events and centralising response.
  Price: for 1-25 units: Manager CHF 7500.-, Server Agent: 3700.-, Workstation Agent: 1500.-.
  Verdict: ITA has potential in a sensitive environment, but is difficult to setup and is expensive. Default configuration for Solaris not good enough.
  It is not easy to get ITA to only report major problems and then report these problems in fine detail.

  Test System: NT4 & Solaris 2.6 agents, Console NT4. Tested V3.0 in Oct'98. According to the CD it was V3.0 SR2 / 16.6.98, but the about box showed V3.0,/12.12.97.

  Overview:
  - ITA has two primary functions:
    1. The logs on several systems can be collected over the network and centrally analysed and alerts raised or action taken, if required.
       On UNIX, syslog, utmp and optionally C2 and utmp logs are examined. Additional ASCII logs can be configured.
       On NT, the System, Application and Security event logs are examined.
    2. The integrity of specified file lists can be checked at regular intervals using Byte Rotary, Word Rotary checksums (what are these?? sound like a waste of time?) and MD5 hashes. The default installation on Solaris includes a short generic list (why isn't /etc/shadow or /etc/default on the list? /etc/passwd has two entries and /etc/inetd.conf is misspelled).
       See also /omniguard/ita/system/THIS_HOSTNAME/ita.ini
       Note: checksums should *never* be used, they are too each to fool. Only strong one way functions such a hashes (MD5, SHA-1) should be be used.
  - An ITA "agent" is installed on each host to be monitored. The agent communicates with a "manager" which collates events. The main GUI runs on "Admin" hosts, which can connect to one or more "managers" and display/configure the manager & it's agents.
  - Policies are defined which stipulate what expressions to monitor/ignore in logs (called *Rule Select/Ignore Clause* s) and take a corresponding action (called *Rule Action Clause).* Default policies are delivered with ITA, these may be copied and adapted or new policies aded.
    - *Rules* have an alert level: Green, Yellow or Red (called a rule value).
    - *Select/Ignore* clauses allow use of the * (any number of char.) and ? (one character) in expressions for matching log entries.
    - Other *select* options: Events generated by particular users or systems, ITA client/server commands/ messages/ errors, flags raised by other events, with a certain date/time period & frequency [the interface is a bit clumsy on this one] and timers (don't understand this one).
    - *Actions* can be diverse: append a message to a local or remote log, send an SMTP email, send a "Windows notification" to a user, call a pager (via a modem), kill a process (if the PID is available on UNIX or the username is available on NT - all processes for that user are stopped), disconnect an session (if Session ID is available in UNIX log, or username for NT), raise/cancel a flag for a certain time, start/cancel a timer, execute a (user defined)

script/command, record an event in the ITA database, disable a (non admin) user account (if the username is available from the logs).
  - Shared Actions can be defined to allow actions to be shared between several rules.
- On Solaris only certain syslog entries are monitored by default. The following entries are added to /etc/syslog.conf on installation:

```
*.info;mail.err;mark.none /omniguard/ita/system/THIS_HOSTNAME/syslog
mail.debug /omniguard/ita/system/THIS_HOSTNAME/syslog.mdbg
```

  The script /omniguard/ita/bin/itarc has to be modified so that ITA_EXTRA_FILES includes a list of logs to monitor. e.g.

```
solaris-sparc)
  ITA_EXTRA_FILES='/var/adm/sulog,/var/adm/loginlog'
;;
```

- ESM logs are automatically monitored.
- Various users with differing rights can be configured in ITA to allow sharing of responsibilities. The rights are: View ITA configuration, Modify Policies & Domains, view Events, change Manager configuration, change Agent configuration, register new Agent, user Account Information.
- ITA View is a tool that allows querying of historical and real-time events. Data can be presented in graphical or tabular form.
  - Generic predefined views are: FYI/Text, Moderate Concern/graph, Moderate Concern/text, Emergency/graph, Emergency/text.
  - Events can be selected by policy & agent in additional to manual queries of variables (User, System, Rule, Policy, Value, Text and user defined variables). User variables are those defined when specify new logs and their formats (Right click on an agent and select properties to see how these logs are added).
- There is an website operated by Axent security specialists (www.axent.com/swat/swat.htm) which should help ITA users keep up to date on the latest attacks and how to detect them. This is potentially a great service.
  - The policy download section: No NT or UNIX polices have changed since the SR2 release (June 1998).
  - The attack signatures section: presents recent attacks and how to recognise them. A signature ID is provided for each attack, but it is password protected. Presumably if one has a Axent maintenance contract you can download policies to detect these attacks. Last updated 22.9.98.
  - The security tasks section: provides FYI policies for gathering information such as user logons. Last updated 3.8.98.
  - The threats section: provides policies for reacting to threats such as administrator logons. Polices are few and are limited to FTP root logins on UNIX. UNIX/NT section last updated 6.4.98.
  - The integration section: provides policies to monitor events generated by other applications/products. The list is very short. UNIX/NT section last updated 31.7.98.
  - A demo sections contains some attack demos that work with ITA.
  - The "hack stories" contains only one.
  - The links to hacker and security sites are useful.

Advantages:
- Seems to be well though out as a framework for integrating the logs of diverse systems into one central view.
- Client/server architecture, supports many operating Systems. Useful GUI.
- Policies can be exported/imported to/from ASCII files.

Disadvantages:
- Complex system. Not trivial to setup. On-line help not detailed enough and not context sensitive.
- NT: no policies are available for monitoring changes to important registry keys.
- UNIX: no policies for monitoring SSH connections are included.
- Binary logs can't be monitored on UNIX.
- Expressions used in Select/Ignore clauses are very limited and do not approach the power of typical regular expression engines such as Perl, Awk or egrep. This makes writing of clauses more clumsy and time consuming.
- No command line interface.
- Does not install in standard locations on NT (in /omniguard instead of Program Files).
- The post-install script allows stopping & starting of the manager. It also allows un/registering of the agent to manger on NT, but not on Solaris (why?).
- ITA Admin has no option for testing all agent connections and reporting a status.
- Changes to items in the policy library are not reflected in "applied" polisiec. They need to be deleted and added from the library again.
- Serious: The log entry that kicks off a rule does not get copied to the action. i.e. if the action is notify a user, write to a log or write to ITA view, the name of the event/ time/ seriousness is reported but not the actual log entry. Weird. Hopefully there's be a workaround ?
- Problems:
  - NT System dll times were reported as having changed, but they were exactly one hour different what what was expected. Perhaps ITA has timezone problems? The test was conducted in GMT.
  - How can logs be monitored if they have named that constantly change? Some log names include the current date and are created fresh each morning (e.g. MS HTTP proxy).
  - ITA Admin crashed when a new policy was created and then deleted without making any changes.
  - ITA view crashed when loading the FYIT generic view from localhost.
  - The SYN flooding policy doesn't allow you to set how many TCP open connections are allowed in what timeframe. So it may go off on heavily used servers.

Recommendations:
- To get the best from ITA: Test it understand it. Carefully decide what you want ITA to do & how. What system administrators

will have what responsibilities? Run a pilot test. Go to an Axent ITA course. Plan the final installation.
  - You need a fast machine for the Admin, it's slow.
  - Check /omniguard/ita/system/THIS_HOSTNAME/agent.log and manager.log to make sure that the agent was able to attach it self to the various logs & the manager started OK.
  - If being used for firewalls, specific sockets will have to be opened up between agent & manager(by default port 5051 is used for Agent->Manager and 5052 for Manager->Agent). The port number for ITA view can be changed. Dynamic ports are also allocated to the manager for client communication. The range of dynamic ports used can be configured in the [Manager] section of ITA.INI. So assuming the manager is on one side of the firewall and the agents on the other:
    - Allow port 5051 from agent to manager.
    - Select a range for client connections, add it to ITA.INI and allow the firewall to pass these ports from manager to agent.
  - The manager must be a highly secured system, at least as well protected as it's strongest agent (to prevent abuse of trust if attacked).
  - Be careful not to load agent's CPU and the network too heavily. Use the (useful) "throttling" feature in ITA to limit the network bandwidth.
- *CyberCop Server* is Network Associates' (www.nai.com) equivalent to Axent ITA (above).
  See www.nai.com/products/security/cybercopsvr/index.asp
  No Solaris version available for download (but apparently it is available on Solaris).
  Can notify Tivoli or Cisco PIX .
  Test Environment: NT SP3, Installed the 1.01 demo (dated 16.3.98). No uninstall! Crashed NT after 30 minutes. Removed from system.
  Verdict: Test aborted (looks too much like betaware).

Personal Firewalls / intrusion detection systems

I've rewritten this section in a new, separate article, please have a look at the Personal Firewalls article.

Penetration testing

Penetration testing allows the actual security level to be compared with that desired. Here a brief overview of penetration methods/issues are presented. Since this material could be misused to actually attack other firewalls, only relatively well known weaknesses are explained, without providing an incentive to hackers (it is hoped!). It is strongly recommended that the preparation stage mentioned below be carefully executed and that the firewall administrator know of the impending penetration testing, even if this distorts the results somewhat.

Preparation
Obtain management approval, prepare a test plan which indicates:

- Objectives, e.g.:
  - Test the implementation of the policy (report on weaknesses, errors)
  - Test the effectiveness firewall policy (in the policy adequate?)
  - Test the organisational effectiveness/response
- Scope: What will and especially what will not be tested: e.g.,
  - other network entry points/bypass possibilities such as remote access (modem) systems (war dialers)
  - weaknesses in the firewall services /operating systems/configuration,
  - logging levels & analysis.
  - Does the administrator notice a break in and how does he react? Does an Intrusion detection policy exist and is it adhered to?
- How far can the testing go?
  - Whether the firewall operation may be disruptive or not (i.e. strength of attack) and whether attacks are allowed from both inside and outside.
  - Whether denial of services attacks are allowed (unlikely).
  - Whether social engineering attacks are allowed.
  - Whether the firewall administrator should be informed in advance (recommended IMHO, if only a few hours beforehand). This is an important decision that requires management approval.
  - Whether source code is to be examined.
  - Whether router, bridge and wiring is to be examined.
- Project plan with deadlines, costs, description of deliverable (report) and responsibilities.
- Get the plan signed, with permission to start the test on a particular day.

Test Stage 1: Indirect information collection
At this stage, the firewall is not approached, so no attack attempt can be detected.

- Use DNS (*e.g. nslookup /dig* , *whois* , *ARIN*) to see what information is published about the network, try to map it.
- Search public archives (Internet) for postings from employees of this domain (security, UNIX and NT newsgroups, email-list archives etc.).
- Examine the target's Internet WWW, ftp servers for information. Examine sites that may provide information about the company (e.g. SEC).
- If it is already clear what kinds of products are in use in the firewall, get the vendor information and search the Internet (or other sources) for information regarding weaknesses of these products.

Test Stage 2: Direct information collection
Now the firewall administrator may detect us, but no disruption of the firewall should occur.

- Check email gateway product name & version.
- Check bounced email headers (send email to non-existent users), to see if the internal machines names/structure is provided.
- Scan (gently, i.e. with ping) address space for available hosts, or do a *stealth scan* , i.e. use a tool which just checks for used ports by sending a TCP FIN packet and waiting to see if an RST is returned (then there is no service), e.g. use nmap.
- Scan phone lines for modems (e.g. with ToneLoc, PhoneSweep or THC)

Test Stage 3a: General attack from outside
The attacks from here on in will make you very popular with the firewall administrator...

- Check for obvious (easy to attack holes), e.g. with *showmount -e* , *rpcinfo* , NT/Win95 Shares, nmap etc. Tools like ISS; Satan/Sara/saint will probably trigger alarms on the firewall and warn the administrator.
- If there are obvious holes at this stage and the holes allow access to machines inside the firewall (i.e. not sacrificial lambs outside), then the advanced techniques are hardly worth the effort.

Test Stage 3b: Advanced techniques from outside

- WWW attack: try to exploit common WWW server vulnerabilities.
- Blind IP spoofing: try to spoof the firewall into believing to are an internal host (which is trusted by the firewall), possibly while blocking the internal host with a SYN flood attack. This will not work if the firewall has a filter that rejects packets from the outside that contain "from" addresses from internal hosts.
- If a server on the same network as the firewall can be penetrated, a more advanced spoof can be attempted in which the output is visible.
    - If the firewall trusts host B, the attacker gains access to host C which is on the same subnet as host B.
    - He can disable host B with a DoS attack for a few seconds and during this time change the TCP configuration of host C to pretend it is host B.
    - Then use the trust to insert a backdoor in the firewall, then stop the spoofing and simply use the backdoor to gain entry.
    - The users/services on host B might not notice the attack, thing the network was overloaded for a few seconds for example.
- If a packet filter is used to only allow incoming connection according to port number, then the source of telnet (for example) can be modified to come from another port and possible be allowed to connect because it is coming from an allowed port.
- Source routing: Most firewalls will discard source routed packets, but it can be tried anyway.
- The high port numbers on Cisco routers can be tried.
- At this stage the firewall configuration needs to be understood to launch an attack on particular services that are available. Since I don't wish this list to be misused, no further information on attacking individual services is provided here.

Test Stage 4: Attack from the inside
Basically all of the methods in Stage 3 are possible, but they are much easier, since the trust of the internal network is generally much higher.

Test Stage 5: Firewall configuration review
Now the auditor comes onsite and completes an on-site audit of the firewall and the network connections.

- Organisation:
    - How are changes made, alerts raised & handled? Does an incident response procedure/policy exist?
    - How many people are responsible? Is responsibility clearly defined?
    - Is Policy clear and correctly implemented? Who defines policy?
    - Are inter-network connections audited? How often? Does a general policy for inter-network connection exist? How is it enforced?
    - If new networks are connected, is the security reviewed on the other side? Is the connection justified business-wise? Who approves connections? Who controls routers?
    - How many people are allowed to configure (have passwords) interface routers and firewalls?
- Technical:
    - Using the knowledge of what exactly the firewall consists of, one can use experience to judge what additional weaknesses to those discovered in the previous stages exist.
    - Checking of all network access points, protocols
    - Network device checking: Hardware & software (OS network services, network applications)
    - Host checking: OS, applications, hidden files, open/changed files, unusual SUID/SGID files, world/group writeable files/directories, hidden/unknown processes, installed compilers/debuggers, logins from unknown hosts/at invalid or unusual times etc.
    - Network vulnerability checking: Remote access points, weaknesses, check for strange packets (incomplete, invalid addresses, source routed etc.)

Report findings
Corrective action: List findings, order by risk, propose corrective action (if required).

Problems:

- Tools are few, incomplete and/or very expensive.
- Information is not always openly discussed (especially as regarding *effective* attack tools). This means that few organisations and hackers have the resources to maintain firewall penetration knowledge. Ideally it should become a science, resulting in better firewall products and awareness, as crash testing of cars is used to decrease risks to human beings.

- Who can you trust to test your firewall?

References: Since the above was written, several net resources can come to light that may help you in penetration testing:

Firewall Piercing mini-HOWTO linuxdocs.org/mini/Firewall-Piercing.html

PEN-TEST Email list on www.securityfocus.com

Default Logins / Passwords: www.nerdnet.com/security/index.php

Hacker Lab: learn how to hack or see how good you are: www.hackerslab.org

Penetration testing WAP:
www.security.nl//misc/infosecurity.nl-2000/ralph_moonen/m-sec.ppt
www.security.nl/misc/infosecurity.nl-2000/ralph_moonen/index.html

URL attacks:
Bruce Schneier, Crypto-Gram, Feb 2001, "A Semantic Attack on URLs" http://www.counterpane.com/crypto-gram-0102.html#7
URL, Little Do We Know Thee, by Razvan Peteanu (URL stale)
http://securityportal.com/articles/urlurl20010307.html
